Module 1 - Sequester Contract Slide 1: Too Many Sources = No Signal

Acknowledge the overwhelming flood of content from multiple channels-social media, forums, video streams-and explain why indiscriminate grazing prevents mastery. Emphasise the need to limit sources to find signal amid noise.

Module 1 - Sequester Contract Slide 2: Single Source of Truth (SST)

Define the concept of a Single Source of Truth (SST): the one authoritative reference that anchors all learning and development. Explain why choosing one curriculum and one reference manual prevents confusion and anchors deep understanding.

Module 1 - Sequester Contract Slide 3: Building the SST-Aligned Workspace

Guide learners through creating a structured workspace that reflects their SST. Suggest using a single notebook or note-taking tool and creating sections for the SST link, personal glossary, weekly learning journal, and a parking lot for unanswered questions-without chasing distractions.

Module 1 - Sequester Contract Slide 4: Declare Your SST

Prompt learners to declare their SST explicitly. Encourage them to write down the title of their chosen reference, link it, and make a pledge to stay with that source until the module concludes. This act of commitment reinforces disciplined learning.

Module 1 - Sequester Contract Slide 5: One Source. One System. One Notebook.

Summarise the sequester contract with a simple mantra: one source of truth, one structured system for working, and one notebook for capturing insights.
Reinforce that constraint is a catalyst for clarity.

Module 2 - Mental Models Slide 1: Chatbot != Agent

Contrast naive chat interfaces with autonomous agents. Explain that while chatbots respond to prompts, agents plan, remember, reason, and act using tools. This sets the stage for deeper mental models.

Module 2 - Mental Models Slide 2: Two Lenses - Teacher-Student & Agent Anatomy

Introduce two powerful mental models: the teacher-student metaphor, where the prompt is the teacher and the model is the student, and the agent anatomy metaphor, with parts like brain (planner), memory, sensors (inputs), actuators (tools), and guardrails.

Module 2 - Mental Models Slide 3: Map Functions to Parts

Demonstrate how to map specific agent functions-planning, memory management, tool invocation, and I/O-onto the mental models introduced. This helps learners attach abstract concepts to familiar structures.

Module 2 - Mental Models Slide 4: Describe Your Agent in Plain Language

Encourage learners to describe what an agent does in simple, clear sentences, avoiding jargon. This exercise ensures they have internalised the mental models and can communicate the agent’s purpose without confusing terminology.

Module 2 - Mental Models Slide 5: Model the Mind Before the Terms

Remind learners to focus on understanding the cognitive structure of agents-the planning, memory, and tool usage-before diving into terminology. The mental model should come first; names and frameworks come second.

Module 3 - Pro-grade Checklist Slide 1: Prompt-and-Pray Fails in Real Work

Expose the limitations of hobbyist approaches that rely solely on prompting. Explain that professional systems require specifications, tests, observability, and guardrails to operate reliably.

Module 3 - Pro-grade Checklist Slide 2: Professional vs Enterprise

Clarify the difference between professional-grade agents and enterprise-grade agents. Professional means well-specified, testable, observable, and documented; enterprise adds scalability, compliance, and operational maturity.

Module 3 - Pro-grade Checklist Slide 3: Grade an Agent Template Against the Checklist

Walk through an example agent template and evaluate it against a professional-grade checklist, highlighting where it passes and where it fails. Show how to improve weak areas.

Module 3 - Pro-grade Checklist Slide 4: Upgrade Plan: From Hobby to Professional

Offer a blueprint for elevating a hobbyist agent to a professional one. Outline steps such as defining a specification, adding tests, implementing guardrails, and instrumenting with tracing and metrics.

Module 3 - Pro-grade Checklist Slide 5: Spec - Test - Trace - Guardrail

Summarise the four pillars of professional agents: clear specifications, comprehensive tests, transparent tracing and observability, and robust guardrails to prevent unintended behaviour.

Module 4 - Deterministic Evals Slide 1: Same Input, Different Output = Chaos

Introduce the problem of non-deterministic behaviour in agents, emphasising how unpredictable outputs undermine trust and hinder debugging.

Module 4 - Deterministic Evals Slide 2: Golden Tests, Fixtures, and Version Pinning

Define golden tests as fixed input/output pairs that agents must reproduce. Explain the role of fixtures and why version pinning is necessary to ensure reproducibility over time.

Module 4 - Deterministic Evals Slide 3: Failing Golden -> Fix -> Re-run

Demonstrate a deterministic evaluation cycle: run a failing golden test, identify the cause, apply a fix, and re-run to confirm the output matches the expected result.

Module 4 - Deterministic Evals Slide 4: Write Your First Golden Test Case

Guide learners through creating their own golden test-choosing a representative input, defining a precise expected output, and storing it for future evaluation.

Module 4 - Deterministic Evals Slide 5: Trust What You Can Re-run

End with a mantra: trust what can be repeatedly re-run with the same output.
Deterministic systems are dependable, debuggable, and scalable.

Module 5 - Design -> Build -> Run Slide 1: What Happens When You Skip Design

Warn learners about the consequences of jumping straight to implementation.
Without a clear design, agents are hard to test, extend, and operate safely.

Module 5 - Design -> Build -> Run Slide 2: Three Lanes: Design, Build, Run

Define the three distinct phases of creating agents: design (specifying scope, constraints, and success metrics), build (constructing prompts, planners, memory, tools, and tests), and run (deploying, monitoring, and iterating).

Module 5 - Design -> Build -> Run Slide 3: Sort a Feature into D/B/R with Hand-offs

Show how to classify tasks or features according to whether they belong in the design, build, or run phase. Emphasise how hand-offs between phases work in practice.

Module 5 - Design -> Build -> Run Slide 4: Classify Six Tasks: D vs B vs R

Present learners with six example tasks and ask them to decide which phase each belongs to. This exercise reinforces their understanding of the roles and responsibilities within an agent project.

Module 5 - Design -> Build -> Run Slide 5: Design First. Build Clean. Run Observed.

Close the module with a succinct mantra encouraging disciplined development:
begin with design, implement cleanly, and operate under observability.